

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.

Plaintiff,

vs.

GOOGLE INC.

Defendants.

Case No. 3:10-cv-03561-WHA

**REPLY EXPERT REPORT OF JOHN C. MITCHELL
REGARDING PATENT INFRINGEMENT**

**SUBMITTED ON BEHALF OF PLAINTIFF
ORACLE AMERICA, INC.**

virtual machine described as an embodiment of the '205 patent when it was written, just as the ability to decode the EXECUTE_INLINE and EXECUTE_INLINE_RANGE instructions was written into the Dalvik VM when it was written.

60. Prof. August fails to consider the evidence that dexopt runs at runtime. As Google wrote: “[t]he solution is to invoke a program called dexopt, which is really just a back door into the VM. It performs an abbreviated VM initialization, loads zero or more DEX files from the bootstrap class path, and then sets about verifying and optimizing whatever it can from the target DEX. On completion, the process exits, freeing all resources.” (*See* <http://android.git.kernel.org/?p=platform/dalvik.git;a=blob;f=docs/dexopt.html>.) The optimizations that dexopt performs (such as replacing symbolic references with vtable offsets) are ones that require information that is only available at runtime (such as a vtable offset), and that information is obtained by initializing a Dalvik VM and loading all the classes from an application’s .dex file into the VM. Google confirms it: “[s]ome of [the optimizations performed by dexopt] require information only available at runtime, others can be inferred statically when certain assumptions are made.” (*See id.*) The way Google wrote dexopt, rewriting the DEX method invocation instructions with EXECUTE_INLINE and EXECUTE_INLINE_RANGE instructions is one of the former kind of optimizations.

2. Infringement by Android’s dexopt under the “JIT Theory”—Android generates new virtual machine instructions

61. As described in paragraphs 38-50 of my Opening Copyright Infringement Report, the Java compiler transforms source code defining a Java class to a class file (containing bytecode) that implements this class. The Java execution platform interprets this intermediate form of executable code in the Java virtual machine. Similarly, Android applications are written in the Java programming language and compiled to Java class files using a Java compiler, that are then converted to .dex files using the Android dx tool. The Dalvik VM executes files in the Dalvik Executable (.dex) format. (*E.g.*, Opening Copyright Infringement Report ¶¶ 81-94.) Prof. August does not dispute this account (*see, e.g.*, August ’205 Report ¶¶ 113 -121) or the fact

X. SUMMARY OF OPINIONS AND CONCLUSION

123. It is my opinion that Google's experts have failed to demonstrate that any asserted claim of the patents-in-suit is not infringed.

124. It is also my opinion that Google's experts have failed to demonstrate that Google is not liable for direct and indirect infringement in the manner described in my Opening Patent Infringement Report.

125. It is further my opinion that Google's experts have failed to demonstrate that the patents-in-suit do not form the basis for consumer demand for Android by developers and end-users.

126. It is also my opinion that Google's experts have failed to demonstrate that the patents-in-suit were not necessary to Android achieving satisfactory performance and security, once Google decided to adopt the Java execution model in Android.

Dated: September 1, 2011



John C. Mitchell